

## **COST ESTIMATION FOR DEVICE TESTING**

### **Background of the Invention**

**[0001]** Prior to shipping a device, such as a system-on-a-chip (SOC), the device must be tested to determine whether it has been manufactured correctly and is fully operational. A variety of testers are available for such testing. Typically, a tester is a very large and expensive machine which is designed to precisely position the placement of logic signal transitions at very high speeds. Most testers are aimed at creating a “functional environment” for a device which mimics the environment in which the device will eventually be used, to thereby demonstrate that the device will behave as expected in that environment.

**[0002]** For functional tests, a tester applies a series of “test vectors” to the inputs of the device. A test vector is a critically timed cycle of events lasting a short period of time referred to as a “vector cycle”. Within a vector cycle, and at precisely calculated times, logic signal drivers in the tester apply stimulus to device inputs. At the same or some precisely delayed time, logic signal comparators in the tester monitor responses at device outputs. When many test vectors are executed sequentially, discrepancies between monitored and expected device outputs, if any, are noted as device failures.

**[0003]** An alternative or adjunct to functional testing is “structural” testing (e.g., “scan” testing). Structural testing enables the testing of structures which are deeply embedded within a device. Rather than testing

the device's internal structure by applying stimulus to the device's inputs, structural testing involves shifting a series of test vectors into the core of a device, and after each test vector is shifted in, launching the test vector and capturing a response. Each response is then shifted out of the device. In this manner, a tester can verify that all of a device's elements are present and operational. An assumption of structural testing is that if all elements are present and operational, then the elements will contribute to performing the greater and intended functions of the device (e.g., adding, shifting, etc.), and the device will function as designed.

**[0004]** Each type of test (functional, structural, or other types of tests) may have different memory requirements for the tester to execute the test vectors for each test to be performed. The requirements may also vary between different tests of each type. Currently, customers or test designers may not know the amount of memory required to execute their tests.

### **Summary of the Invention**

**[0005]** Methods and systems for estimating a cost to execute test vectors are disclosed. In one embodiment, the method comprises reading a test file having a plurality of test vectors. Next, a required memory needed to execute the plurality of test vectors is determined. The required memory is then used to estimate a cost to execute the test vectors.

**Brief Description of the Drawings**

**[0006]** Illustrative embodiments of the invention are illustrated in the drawings in which:

**[0007]** FIG. 1 illustrates an exemplary view of a system which may be used to determine a cost for device testing;

**[0008]** FIG. 2 illustrates an exemplary method that may be used by the system of FIG. 1 to estimate a cost to execute test vectors; and

**[0009]** FIG. 3 illustrates an exemplary method for determining memory needed to execute test vectors for each pin of a tester that may be used by the method of FIG. 2.

### **Detailed Description**

**[0010]** An exemplary embodiment of a system that may be used to estimate a cost to execute test vectors is illustrated in FIG. 1. The system 100 includes logic 102. As will be described in further detail below with reference to FIG. 2, logic 102 may be used to read a test file having a plurality of test vectors and to determine a required memory needed to execute the plurality of test vectors.

**[0011]** The system 100 also includes billing predictor 104 communicatively coupled to logic 102. The required memory determined by logic 102 is used by billing predictor 104 to estimate the cost to execute the test vectors. Cost estimating will also be described in further detail with reference to FIG. 2. In one embodiment, logic 102 and/or billing predictor 104 may be part of a design-to-test tool that is used to generate tests that can be read by a tester during device testing.

**[0012]** System 100 may additionally include user interface 106 communicatively coupled to billing predictor 104 and/or logic 102. User interface 106 may be used to display the cost to a user, such as a test designer. The user may then use the cost information to revise tests to meet a test budget. In one embodiment, costs may be displayed for multiple suppliers of test services and the user may use the information to choose a supplier to perform the tests. It should be appreciated that in alternate embodiments, system 100 may not include user interface 106.

**[0013]** In the configuration described above, different components were described as being communicatively coupled to other components. A communicative coupling is a coupling that allows communication between the

components. This may be by means of a bus, cable, network, wireless mechanism, program code call (e.g., modular or procedural call) or other mechanism that allows communication between the components. Thus, it should be appreciated that logic 102, billing predictor 104, and user interface 106 may reside on the same or different physical devices. By way of example, user interface 106 may be a web browser on a remote client. It should also be appreciated that logic 102, billing predictor 104, and user interface 106 may be implemented in software, firmware, hardware or a combination of these.

**[0014]** As shown in FIG. 2, logic 102 may be used to read 200 a test file containing one or more tests to be performed on a device, such as a system-on-a-chip (SOC). Each of the tests may include a plurality of test vectors to be applied to the device under test. Logic 102 may then determine 205 a required memory needed to execute the plurality of test vectors. By way of example, the number of test vectors for each test in the test file may be counted and the required memory may be determined to be equal to the number of test vectors required for the test with the highest number of test vectors.

**[0015]** The determination of the required memory may vary depending upon the configuration of the tester. In one embodiment, the tester that is used to test the device may include a plurality of boards. Each board may include a plurality of pins that may be used to drive inputs and receive outputs from the device under test. Each pin may include its own memory to use during the testing of the device. The memory may be used to store pin specific vector information. In alternate embodiments, memory may not be

included on each pin, but may instead be included for each board or other component of the tester, or pooled in a central location and dynamically allocated by a centralized test processor.

**[0016]** In one embodiment, logic 102 may determine 205 an amount of pooled memory required for the tester to execute the test vectors. In another embodiment, a required memory needed for each board of a tester to execute the test vectors for the board may be determined. By way of example, in embodiments having a memory associated with each pin, a required memory needed for each board may be determined by determining the memory requirements for the pin with the highest memory usage. In a third embodiment, logic 102 may determine 205 the required memory needed for each pin to execute the vectors for the pin. Appropriate determinations 205 for other configurations are also contemplated.

**[0017]** Billing predictor 104 may use the required memory determined 205 by logic 102 to estimate a cost to execute the test vectors. The cost may be estimated by using a billing rate and billing scheme charged for the use of various amounts of memory. The billing rate/scheme may correlate to rates charged by a supplier of test services or may be rates charged to use memory of a tester. Additional factors, such as speed requirements, may also be taken into account when estimating the cost.

**[0018]** FIG. 3 illustrates an exemplary embodiment of a method for determining 205 a required memory that may be used by logic 102. The method begins by determining 305 a first memory requirement for a first pin to execute the test vectors for a first test in the test file. By way of example, the first memory requirement may be determined 305 by counting the number of

test vectors in the first test for the first pin. The required memory is then set 310 to be equal to the first memory requirement.

**[0019]** Another pin of the tester having test vectors in the first test is selected and a second memory requirement for the selected pin to execute the test vectors for the first test is determined 315. The second memory requirement may be determined 315 by counting the number of test vectors in the first test for the selected pin. If the second memory requirement exceeds the current value of the required memory 320, the required memory is set 325 equal to the second memory requirement.

**[0020]** After 325, or if the second memory requirement does not exceed the current value of the required memory 320, a determination is made as to whether there are more pins 330 having test vectors in the first test to process. If there are more pins, processing continues back at 315 for the next pin. Otherwise, a determination is made as to whether there are more tests in the test file 335.

**[0021]** If there are more tests, 315-330 are repeated for the next test for each pin having test vectors to execute for the test. After all the tests have been processed, the method ends 340. Thus, it should be appreciated that at the conclusion of the method the required memory is determined to be equal to the memory requirements for the test and pin combination with the highest memory requirements.

**[0022]** In alternate embodiments, the required memory may be determined in a manner different from that shown in FIG. 3. The determination 205 may depend upon how available memory is allocated in the tester. For example, in one embodiment, the memory available for a pin may

depend on the board where the pin is located. Pins on one board may have the same amount of memory available as other pins on the same board, while the amount of memory available for pins may vary between boards. In this embodiment, a required memory may be calculated for each board by determining the memory requirements for the test and pin combination with the highest memory requirements for each board using a method similar to that described in FIG. 3. In a second embodiment, memory may be allocated on a per pin basis. In this embodiment, a required memory may be determined for each pin. Other exemplary embodiments, such as embodiments with one memory available for all the pins of a board, may use corresponding different calculations.

**[0023]** The billing predictor 104 may use various billing rates and schemes that correlate to the configuration of the tester to estimate the cost required to execute the tests. By way of example, in a per pin configuration, billing predictor may use a rate charged for memory to calculate costs to use the required memory for each pin and then total the costs. Correspondingly different calculations may be made for different test environments, such as one memory for tester, or configuration in which all pins on a board are given the same amount of memory but memory can vary between boards.

**[0024]** The cost estimated 205 by billing predictor 104 may then be displayed to a user. Additional information, such as the test requiring the most memory, may also be provided. The user may use this information to find the lowest cost supplier of test services and/or to reconfigure their tests to meet a test budget.



**[0025]** While illustrative and presently preferred embodiments of the invention have been described in detail herein, it is to be understood that the inventive concepts may be otherwise variously embodied and employed, and that the appended claims are intended to be construed to include such variations, except as limited by the prior art. It should also be appreciated that the methods described above may be performed by hardware components or may be embodied in sequences of machine-executable instructions, which may be used to cause a machine, such as a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the actions set forth in FIGS. 2 and 3. Alternatively, the methods may be performed by a combination of software, firmware, and hardware.